Electronic Communications of the EASST
Volume 080 (2021)

Conference on Networked Systems 2021 (NetSys 2021)

Towards Optimization-Based Predictive Congestion Control for the Tor Network

Christoph Döpmann, Felix Fiedler, Sergio Lucia and Florian Tschorsch

4 pages

Guest Editors: Andreas Blenk, Mathias Fischer, Stefan Fischer, Horst Hellbrueck, Oliver Hohlfeld, Andreas Kassler, Koojana Kuladinithi, Winfried Lamersdorf, Olaf Landsiedel, Andreas Timm-Giel, Alexey Vinel

ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Towards Optimization-Based Predictive Congestion Control for the Tor Network

Christoph Döpmann, Felix Fiedler, Sergio Lucia and Florian Tschorsch

TU Berlin and TU Dortmund

Abstract: The multi-hop nature of the Tor network makes it challenging to achieve efficient and fair congestion control. In this paper, we investigate PredicTor, a novel approach towards multi-hop congestion control based on distributed model predictive control (MPC), an advanced optimization-based control technique.

Keywords: multi-hop congestion control, Tor, distributed model predictive control

1 Introduction

The Tor network [DMS04] serves a globally growing demand for online privacy, supporting freedom of speech and censorship circumvention. This growing demand comes with higher expectations with respect to the quality of service, i.e., bandwidth, latency, and fairness. In Tor, congestion control is a major factor for suboptimal—sometimes poor—performance [AG16].

In order to tackle the performance problems, we build upon and extend our previous work on *PredicTor* [FDTL20]. PredicTor is the conceptual study of a congestion control mechanism for the Tor network that builds upon advanced optimization-based control techniques. In particular, it operates *predictively* using *distributed model predictive control*. In [FDTL20], we presented the concept behind PredicTor as well as the optimization problem's formal description.

In this paper, we complement a networking perspective and investigate PredicTor's ability to improve congestion control in non-trivial network scenarios. More specifically, we make the following novel contributions: First, we elaborate on the networking protocol of PredicTor and present thoughts on its implementation. Second, we carry out an initial simulation-based evaluation of PredicTor in networks with complex topologies. We thereby demonstrate PredicTor's potential to clearly improve latency and fairness in multi-hop scenarios. We also point out current weaknesses, leading to our third main contribution: We make the idea of predictive congestion control accessible to the networking community, opening a perspective on future research directions for congestion control, especially in multi-hop scenarios. Our work points out open research questions that will have to be tackled in order to follow this promising path.

2 PredicTor

The Tor network [DMS04] implements the *onion routing* protocol. That is, user data is forwarded over a series of intermediate servers (relays) along a cryptographically-secured path (circuit). This *multi-hop* data transfer constitutes a challenge towards usable performance. In particular, proper congestion control is challenging, because a circuit consists of several independent TCP connections in a sequence. As a consequence, Tor currently fails to deliver low latency as well as fair network resource allocation for circuits [AG16].

To tackle this issue, we introduced the concept of PredicTor in [FDTL20]. PredicTor is built on the idea of *distributed* model predictive control (MPC), a novel technique from the field of control theory. In contrast to traditional approaches, MPC yields the potential to clearly outperform today's congestion control due to its predictive behavior. It relies on a mathematical system description (model) and a formal optimization objective (cost function). Furthermore, it allows the flexible integration of constraints to be satisfied. The resulting optimal control problem is solved repeatedly using numerical optimization. For PredicTor, we encoded the following goals into this mathematical framework to define the optimal scheduling strategy for any point in time:

Max-min fairness The resource allocation between circuits should adhere to local max-min fairness which is known to converge to global fairness.

Minimize queue sizes By taking the current and future buffer sizes into account, the scheduler can proactively prevent congestion in the network. Minimizing queues results in lower latency.

Maximize data throughput At the same time, the optimization goal consists in allocating the maximum possible data rates to circuits such that the available bandwidth is fully utilized.

We refer to [FDTL20] for the full description of the resulting formal model.

3 Network Protocol Details

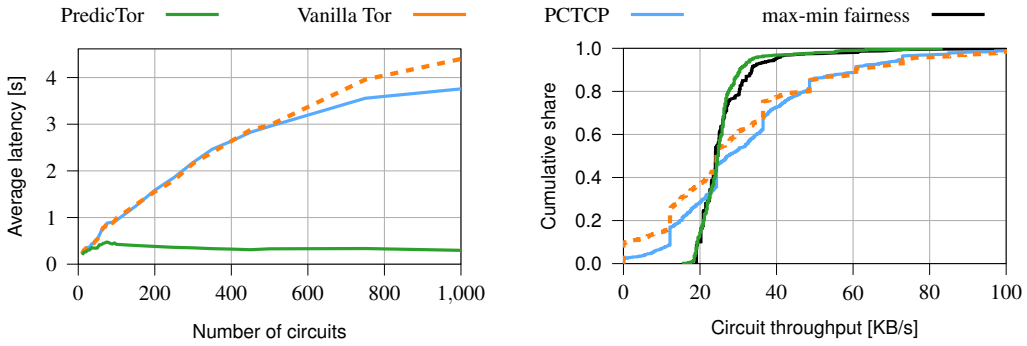
It lies in the nature of distributed MPC that multiple entities interact so they can react to each other's behavior. In the case of PredicTor, each Tor relay independently carries out the optimization locally to define the scheduling strategy that is optimal at this moment in time. The predicted behavior is shared with the relay's neighbors so it can be taken into account for their upcoming optimization steps. These predictions are represented by numeric vectors, so-called *trajectories*, that contain the future expected data rates and buffer sizes. We call the messages used for exchanging them *feedback messages* because they create the feedback loop between any two adjacent Tor relays as far as congestion control is concerned.

Feedback messages carry the following trajectories: the expected input and output data rates, `VIn` and `VOut`, a desired output data rate `VHatOut`,[1] and the expected development of local buffer sizes `SBuffer`. Each trajectory consists of ten double values and a one-byte type identifier. These trajectories are included *per shared circuit* between two adjacent relays.

We envision the exchange of feedback messages going hand in hand with a novel transport protocol for Tor, avoiding issues that stem from the use of TCP [AG16]. Using, e.g., UDP gives PredicTor maximum flexibility for scheduling, reliability, and flow control on the application layer. Currently, Tor sends data as fixed-size *cells* over TCP. Similarly, PredicTor splits its feedback messages into equally-sized chunks to mitigate inference of, e.g., the number of circuits.

Note that PredicTor (same as Tor) exchanges control messages hop-by-hop only, even though the propagation of optimization results leads to distributed convergence towards a network-wide scheduling strategy. We argue that the changes are compliant with Tor's adversary model as information (data rates and buffer sizes) is exchanged between neighbors only. However, a more comprehensive security analysis would be desirable for the future.

[1] This is a minor improvement of PredicTor w.r.t. robustness, which we cannot elaborate due to space constraints.



(a) **Latency** (50 relays, variable circuits)

(b) **Fairness** (50 relays, 750 circuits)

Figure 1: Evaluation results. (1a) Average byte-wise latency (different random scenarios, 25-fold repetition each), and (1b) fairness (CDF plot of throughput per circuit, representative example).

4 Evaluation

In [FDTL20], PredicTor was evaluated only for trivial model scenarios to visualize its general behavior. We continue this work and give an initial idea of how PredicTor performs in non-trivial networks. In particular, we include considerably larger, random networks and compare PredicTor's performance against (vanilla) Tor as well as PCTCP [AG13], an alternative circuit handling strategy for Tor.

To this end, we implemented a proof of concept for the ns-3 network simulator[2], which has a series of limitations. However, on this conceptual level, our main objective is to explore the potential of distributed MPC for congestion control.

4.1 Scope and Limitations

One of our main simplifications includes that we do not simulate the transmission of feedback messages, but emulate their exchange "out of band". Since feedback traffic is independent of the network speed, running the simulations at different simulated data rates, one could achieve arbitrary goodput-overhead ratios, which we refrain from doing. However, we note that, in real networks, feedback overhead would constitute a severe issue due to its linear growth with the number of circuits. Another limitation is that our simulated network exhibits constant, uniform latency between relays. However, the mathematical PredicTor system model can be extended to handle varying latencies [FDTL20]. Moreover, for this prototype, we base the data transfer on TCP (like Tor and PCTCP) instead of introducing a tailored transport protocol. PredicTor thus merely takes the role of a scheduler.

4.2 Methodology and Results

We generate simulation scenarios with 50 relays and a varying number of circuits, ranging from 10 to 1,000. Circuits consist of three randomly selected relays, each requesting an infinite stream

[2] https://www.nsnam.org/

of data (bulk traffic). After a lead time, we evaluate the steady state (identified manually by ourselves), i.e., the last two seconds of simulation time. For this time span, we analyze the following metrics: byte-wise end-to-end latency, throughput, and fairness.

Figure 1 presents our results for achieved latency and fairness. With respect to latency, we can see that PredicTor offers great potential to heavily improve on the status quo (see Figure 1a). In particular, by explicitly requiring small queues during optimization, PredicTor achieves low latency independently of the number of circuits. In contrast, latency grows indefinitely for denser networks in the case of vanilla Tor and PCTCP. At the same time, we observe unfair data rate allocations for vanilla Tor and PCTCP when compared to an optimal max-min fair rate distribution (see Figure 1b): Most circuits receive either too low or too high data rates than their fair share. In PredicTor, however, circuits achieve data rates very close to max-min fairness. One can also see that PredicTor tends to give circuits rather too low data rates than too high. This also becomes apparent when analyzing the overall achieved throughput (not depicted for brevity): PredicTor consistently achieves around 20% less throughput than vanilla Tor and PCTCP. This was to be expected due to the fact that our PredicTor prototype operates as an additional scheduler limiting data rates on top of TCP.

All in all, one can say that PredicTor illustrates the strong potential of distributed MPC for (multi-hop) congestion control, due to its effectiveness in improving latency and fairness. However, several issues like linearly growing feedback overhead, adaption to dynamic latencies, and slightly degraded throughput remain as open research questions.

5 Conclusion

In this paper, we presented a networking perspective on PredicTor, a multi-hop congestion control approach based on distributed MPC. Our results show that MPC bears the potential to clearly improve latency and fairness. Open research questions include reducing the overhead of feedback messages, which currently grow linearly with the number of circuits.

References

[AG13] M. AlSabah, I. Goldberg. PCTCP: Per-Circuit TCP-over-IPsec Transport for Anonymous Communication Overlay Networks. In *CCS '13: Proceedings of the 20th ACM Conference on Computer and Communications Security*. 2013.

[AG16] M. AlSabah, I. Goldberg. Performance and Security Improvements for Tor: A Survey. *ACM Computing Surveys* 49(2), 2016.

[DMS04] R. Dingledine, N. Mathewson, P. F. Syverson. Tor: The Second-Generation Onion Router. In *USENIX Security '04: Proceedings of the 13th USENIX Security Symposium*. 2004.

[FDTL20] F. Fiedler, C. Döpmann, F. Tschorsch, S. Lucia. PredicTor: Predictive Congestion Control for the Tor Network. In *CCTA '20: Proceedings of the 2020 IEEE Conference on Control Technology and Applications*. 2020.